An Introduction to Identity Based Encryption

Matt Franklin

U. C. Davis

NIST Workshop, 3-4 June 2008

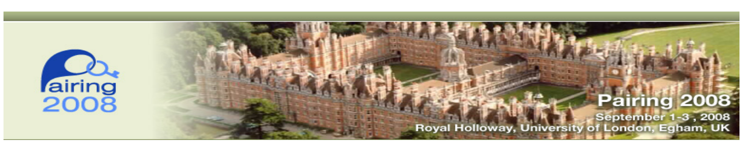
Pairings in Cryptography

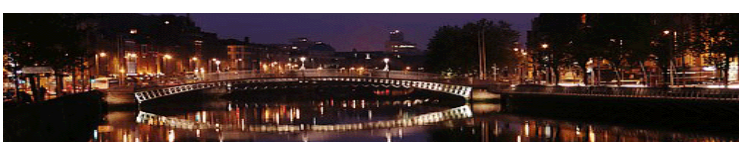
- Tool for building public key primitives
  - new functionality
  - improved efficiency
- Identity Based Encryption [BF2001]
  - early pairing-based construction
  - 1700 citations to date (Google Scholar)

Pairings: Extra Structure on Elliptic Curves

- A. Weil 1946: Pairings defined
- Miller 1984: Algorithm for computing
- MOV 1993: Attack certain elliptic curve crypto
- 2000-today: Lots of crypto applications
  – Joux 2000, Sakai-Ohgishi-Kasahara 2000

Conferences and Workshops in Pairing-Based Cryptography





2005 International Workshop on Pairings in Cryptography (Dublin)

Commercial Interest in Identity Based Encryption

- Mitsubishi, Noretech, Trend Micro, Voltage
- IBE in Smartcards
  - HP/ST Microelectronics, Gemplus
- IBE in email implementations
  - Network Solutions, Microsoft, Proofpoint, Code Green Networks, NTT, Canon, …

Standards Interest in Identity Based Encryption

- IEEE 1363.3 working group: “Identity-Based Cryptographic Methods using Pairings”
- IETF S/MIME working group

Today's Talk:

- Identity-Based Encryption
  - – Functionality and Motivation
  - – Models and definitions
  - – Constructions
  - – Applications
  - – Conclusions

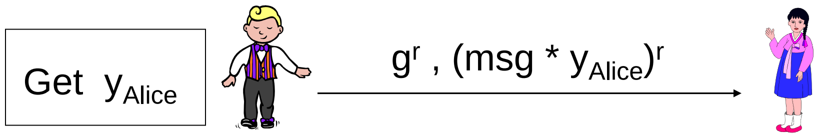
Recall: Public-Key Encryption

Get PK_{Alice}

$E(PK_{Alice}\ ,\ msg)$ $\longrightarrow$

$G(\lambda) \rightarrow PK, SK$	output pub-key, secret-key
$E(PK, m) \rightarrow c$	encrypt message using pub-key
$D(SK, c) \rightarrow m$	decrypt ciphertext using secret-key

ElGamal Public-Key Encryption



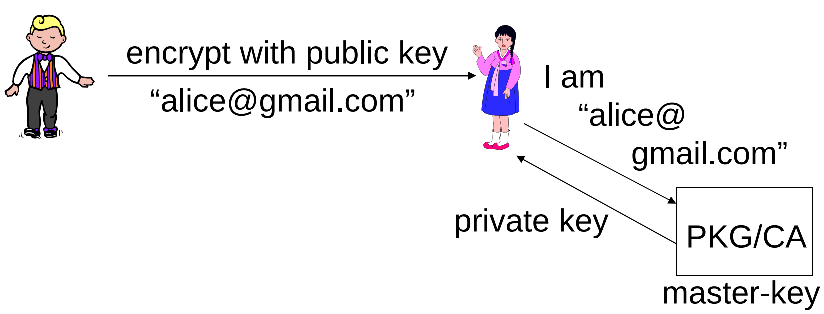
$G(\lambda) \rightarrow PK = (G, g, q, y = g^x), SK = x$

$E(PK, m) \rightarrow c = g^r, (m * y^r)$

$D(SK, c) \rightarrow m = (m * y^r)/ (g^r)^x$

Identity Based Encryption [Sha 1984]

public-key encryption scheme
where PK is an **arbitrary** string



Identity Based Encryption

$S(\lambda) \rightarrow PP, MK$ — output params, master-key

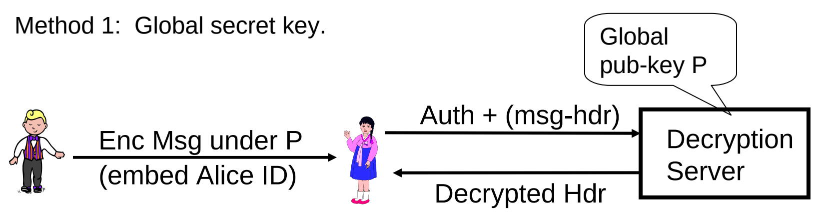
$K(MK, ID) \rightarrow d_{ID}$ — output private key for arb string

$E(PP, ID, m) \rightarrow c$ — encrypt using pub-key, params

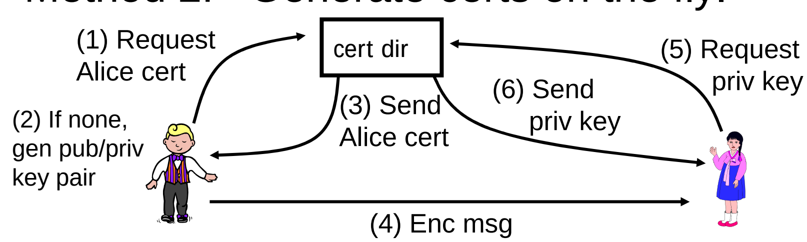
$D(d_{ID}, c) \rightarrow m$ — decrypt using private key

IBE-Like Functionality from Public Key Encryption

- Method 1: Global secret key.



- Method 2: Generate certs on the fly.

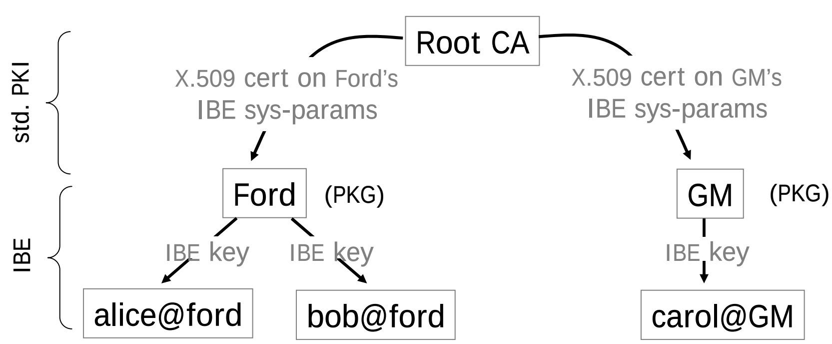


IBE Secure Email

- pub-key “alice@gmail.com”
  - No need to look up Alice’s cert (just params)
- pub-key “alice@gmail.com, current-date”
  - Short-lived (ephemeral) private keys
  - No CRL’s for revocation
- pub-key “alice@gmail.com, date, project”
  - User credentials embedded in public key
  - User credentials managed at PKG/CA

Hybrid PKI

- IBE at user level. Standard PKI at org. level.



alice@ford ⇒ carol@GM :
1. obtain GM’s cert, 2. encrypt to carol@GM

Not Easy to Build IBE

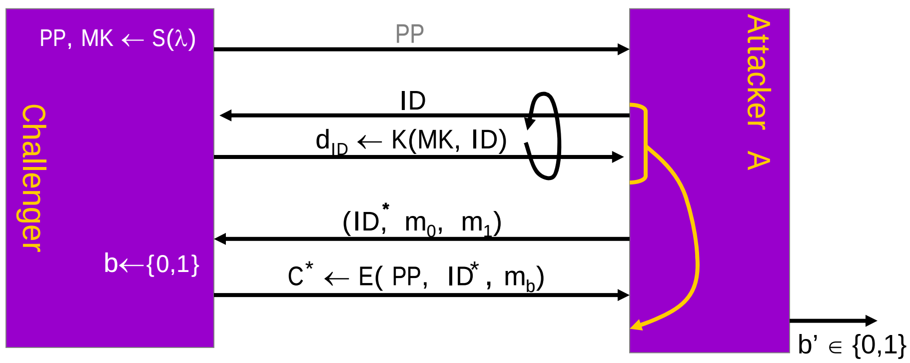
- from ElGamal?
  - Could have params = G, g, q
  - Could map arbitrary ID to ElGamal pub-key y
  - Can’t compute private key for y (DLog)
- from RSA?
  - Can’t map arbitrary ID to RSA modulus N = pq
  - Can’t have common modulus N = pq in params

BF-IBE [Crypto 2001]

- Practical pairing-based IBE
- Performance (courtesy Ben Lynn, PBC)
  - 1 GhZ P3, 1024-bit Dlog security
  - Key generation time: 3 ms.
  - Ciphertext size: 170 bits + ||msg||
  - Encrypt/decrypt time: 19 ms.

IBE Security (IND-IDCPA) [BF'01]

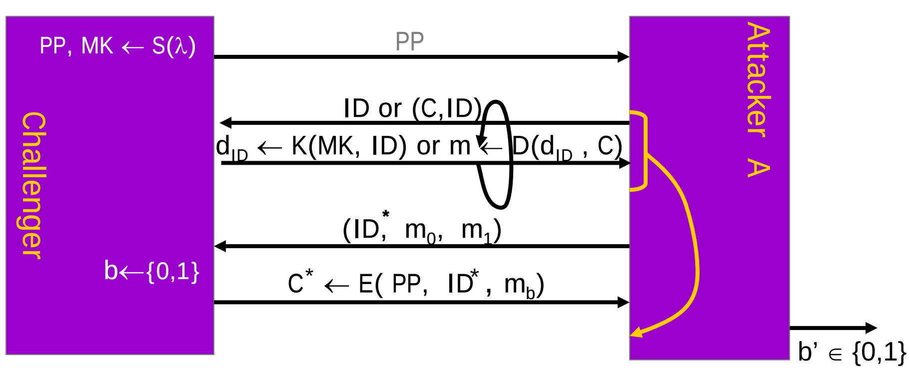
- attacker can request private keys



(S,K,E,D) is IND-IDCPA secure if $\forall$ PPT A: $\left|\Pr[b=b'] - \frac{1}{2}\right| < neg(\lambda)$

IBE Security (IND-IDCCA) [BF'01]

- attacker can request private keys + decrypts



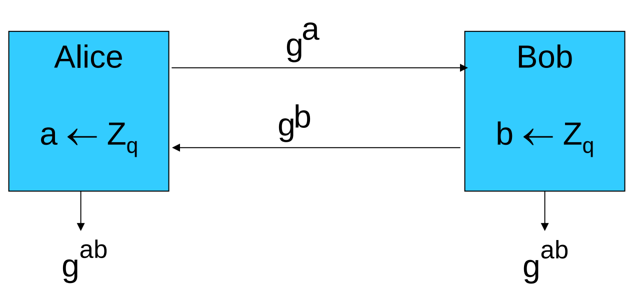
(S,K,E,D) is IND-IDCCA secure if $\forall$ PPT A: $\left|\Pr[b=b'] - \frac{1}{2}\right| < neg(\lambda)$

Security of BF-IBE

- BF-IBE is IND-ID-CCA secure in the random oracle model assuming the hardness of “Bilinear Diffie Hellman”
  – pairings analogue of traditional Diffie Hellman

Recall: Traditional Diffie-Hellman

- G: group of prime order q
- $g \in G$ generator



Traditional Hardness Assumptions

- Computational Diffie-Hellman:

$$g,\ g^x,\ g^y \ \Rightarrow\ g^{xy}$$

- Decision Diffie-Hellman:

$$g,\ g^x,\ g^y,\ g^z \ \Rightarrow\ \begin{cases} 0 & \text{if } z=xy \\ 1 & \text{otherwise} \end{cases}$$

- Discrete-log: $g,\ g^x \ \Rightarrow\ x$

Traditional Hardness Assumptions

CDH, DDH, Dlog believed hard in groups:

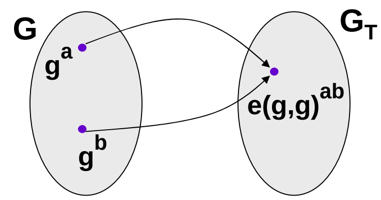
$(Z/pZ)^*$ for prime p

Elliptic Curves $E(\mathbf{F}_p)$: $y^2 = x^3 + ax + b$

	Dlog Alg	Time
$E(\mathbf{F}_p)$	Pollard Rho	$\sqrt{\mathbf{p}}$
$(Z/pZ)^*$	GNFS	$\approx \mathbf{e}^{\sqrt[3]{\ln p}}$

Pairings

G, G_T finite cyclic groups of prime order q



$e: G \times G \to G_T$ is efficiently computable, <u>bilinear</u>, and <u>non-degenerate</u>.

$e(g^x, h^y) = e(g^y, h^x)$

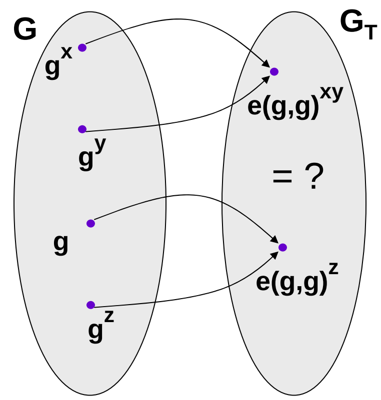
if g generates G, then $e(g,g)$ generates G_T

Bilinear Groups

- G is a "bilinear group" if:
  - $e: G \times G \to G_T$ is a pairing:
    - efficiently computable, bilinear, non-degenerate.
  - G, G_T cyclic groups of prime order
  - Efficient group operations in G, G_T
    - Compact representation of elements of G, G_T
- A number of suitable constructions

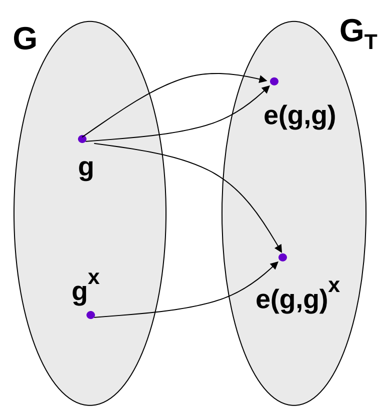
Consequences of Pairings

DDH in G is easy
[Joux 2000, JN2001]



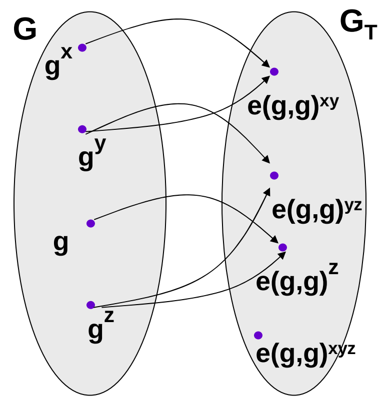
Consequences of Pairings

DLog reduction from G to G_T
[MOV1993]



Bilinear Diffie Hellman

Find $e(g,g)^{xyz}$ in G_T
from g, g^x, g^y, g^z in G



BF-IBE Details [P1363.3 draft]

$S(\lambda) \rightarrow PP = (G, G_T, e, g, g^{\omega})$, and
$MK = \omega$ random in Z_q.

$H_1: \{0,1\}^* \rightarrow G$, $H_2: G_T \rightarrow \{0,1\}^{|m|}$,
$H_3: \{0,1\}^{|m|} \times \{0,1\}^{|m|} \rightarrow Z_q$, $H_4 : \{0,1\}^{|m|} \rightarrow \{0,1\}^{|m|}$

$K(MK, ID) \rightarrow d_{ID} = H_1(ID)^{\omega}$

$E(PP, ID, m) \rightarrow c = (g^r, s \oplus H_2(e(H_1(ID), g^{\omega})^r), m \oplus H_4(s))$
for $r = H_3(s,m)$, s random in $\{0,1\}^{|m|}$

$D(d_{ID}, (u,v,w)) \rightarrow m = w \oplus H_4(v \oplus H_2(e(u, d_{ID})))$, but
reject unless $g^r = u$, for $r = H_3(v \oplus H_2(e(u, d_{ID})), m)$

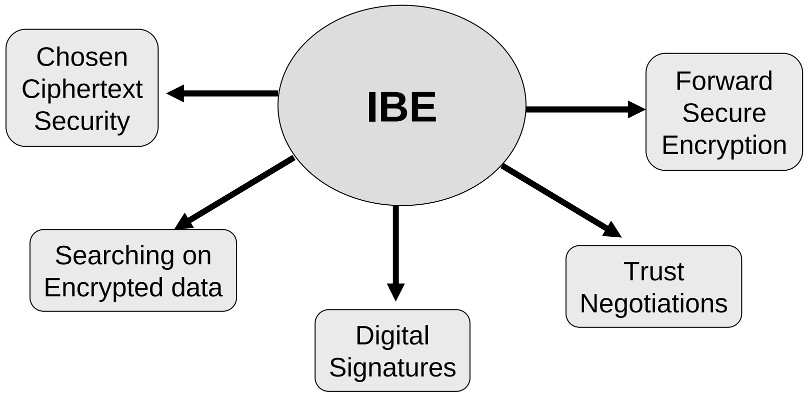
Pairing-Based Cryptanalysis

- Worldwide effort, many researchers
  - Satoh, Shparlinski, Galbraith, Koblitz, Menezes, ...
- No attacks on core hardness assumption
  - Bilinear Diffie Hellman
- No significant attacks on BF-IBE

Other IBE Constructions

- Pairing-Based
  - Boneh, Boyen (BB1) [2004]
  - Waters [2005]
- QR-Based
  - Cocks [2001]
  - Boneh, Gentry, Hamburg [2007]
- Lattice-Based
  - Gentry, Peikert, Vaikuntanathan [2008]

Other IBE Applications



Signatures from IBE [Naor 2001]

private key … master-key MK
public key … params PP
sign msg … private key d_{msg}

verify sig … $E(PP, ID = msg, m) \to c$,
$D(d_{msg}, c) \to m$ for arb m

If IBE is IND-ID-CPA secure, then signature scheme is GMR-secure (strong unforgeability).

Simple Bilinear Signatures [BLS 2001]

Hash H: $\{0,1\}^* \rightarrow G$, $g \in G$, $|G|=q$

<u>KeyGen(</u>λ): $\alpha \leftarrow Z_q$, $y \leftarrow g^\alpha$

<u>Sign</u>(α, m) = $H(m)^\alpha$

<u>Verify(</u>y,m,sig): $e(sig, g) \stackrel{?}{=} e(H(m), y)$

$$\underbrace{e(sig, g)}_{= e(H(m)^\alpha,\, g)} \quad \underbrace{e(H(m), y)}_{= e(H(m),\, g^\alpha)}$$

Security of BLS Signatures

- BLS signature scheme is GMR-secure (strongly unforgeable) in the random oracle model assuming the hardness of Computational Diffie Hellman in G:
  - find g^{xy} from g, g^x, g^y in G (bilinear group).

Properties of BLS Signatures

short

BLS	RSA	DSS
160 (bits)	**1024** (bits)	**320** (bits)

aggregatable

User 1: PK_1 , m_1 $\rightarrow$ S_1

User 2: PK_2 , m_2 $\rightarrow$ S_2

⋮

User n: PK_n , m_n $\rightarrow$ S_n

S

Conclusion

- Identity Based Encryption
  - public key can be an arbitrary string
  - simplifies management of public keys
    - Reduced need for user-level certificate directory
    - Especially well suited for ephemeral public keys
- Pairings in Cryptography
  - Many other applications
  - Revolutionizing public key crypto